Lab 7 Report

ECE 332 Winter 2020

Jose Manuel Lopez Alcala & Benjamin Carlson

1 Introduction

In this lab, we went learned how to characterize a DC motor using a first-order equation. This lab is very similar to Lab 6, but the only difference is that the motor is a DC motor instead of an induction motor. We had to characterize a motor, tune a PID loop, and then implement the PID loop instead of our discrete controller that we design first.

2 Methods

2.1 Hardware

The DC motor was connected to the driver board using two wires for the positive and negative terminals. In addition to that, the motor had the encoder connected to the controlling system so that we could see the output of the motor on our screens. Figure 1 shows the board to which we connected the motor to.

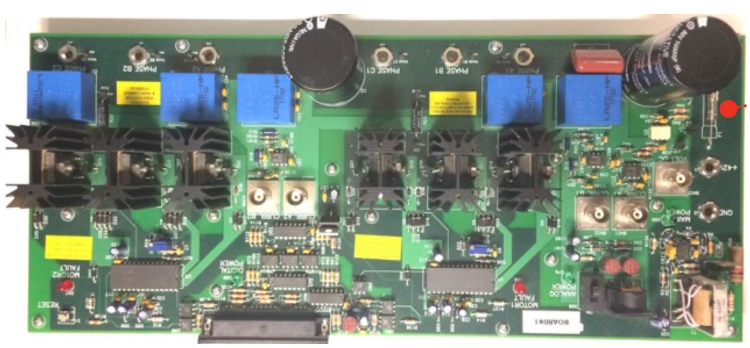

Figure 1. Controlling board for motor

The board was connected to the power supply that supplied the motor power. Additionally, it had another power supply that powered the logic on the board itself. This power supply is shown in Figure 2.

Figure 2. Power supply for logic on controlling board.

The DC motor was powered by the motor driver through an active bridge inverter on the board. The setup can be seen in Figure 3.

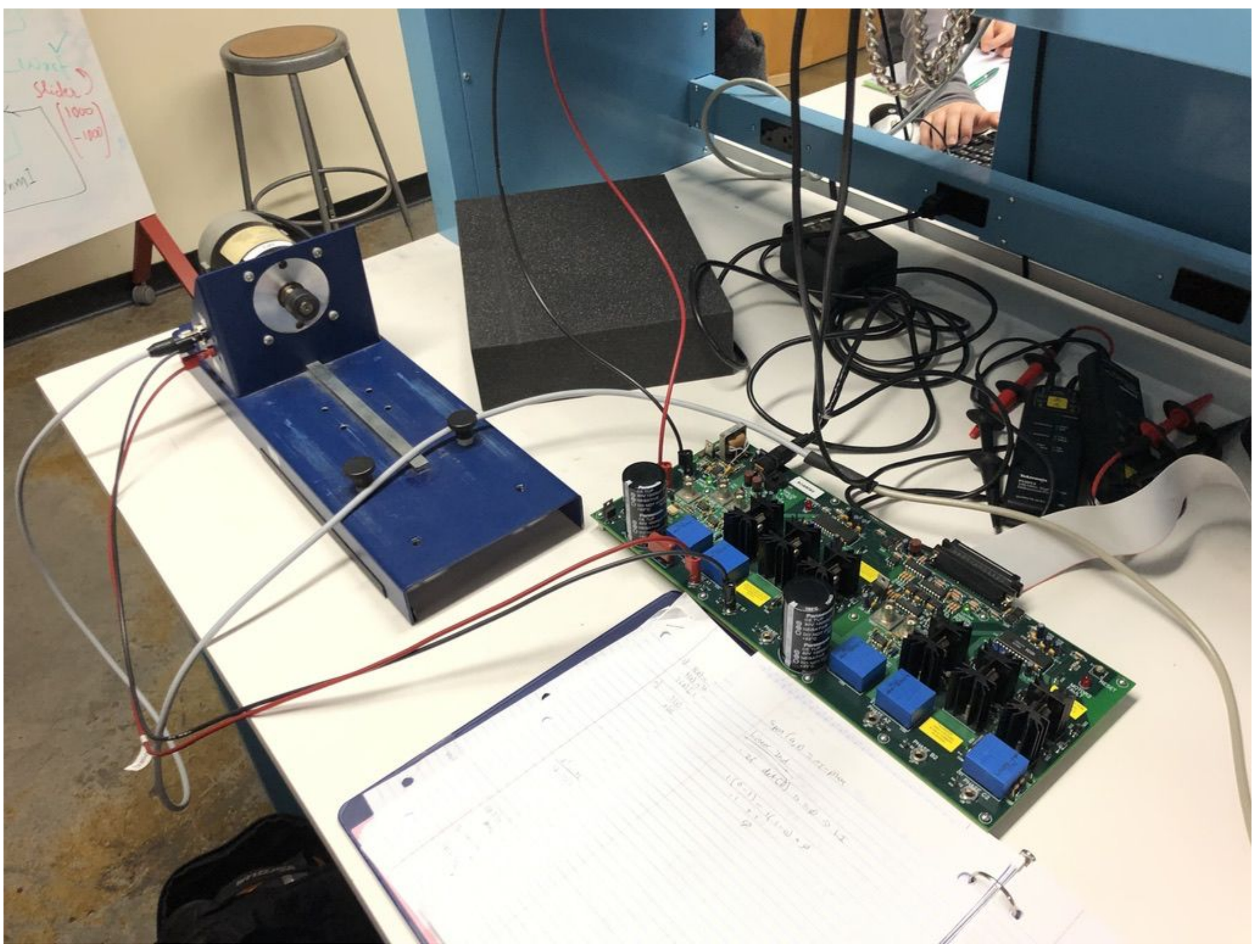

Figure 3. Experimental set-up.

2.3 Motor Characterization

The motor’s response to a step input must be sampled using an open-loop controller. This means that there is no compensation in the output of the motor response. This ensures that the output is the raw, uncompensated value. Figure 4 shows a theoretical output from an uncompensated output of the induction motor.

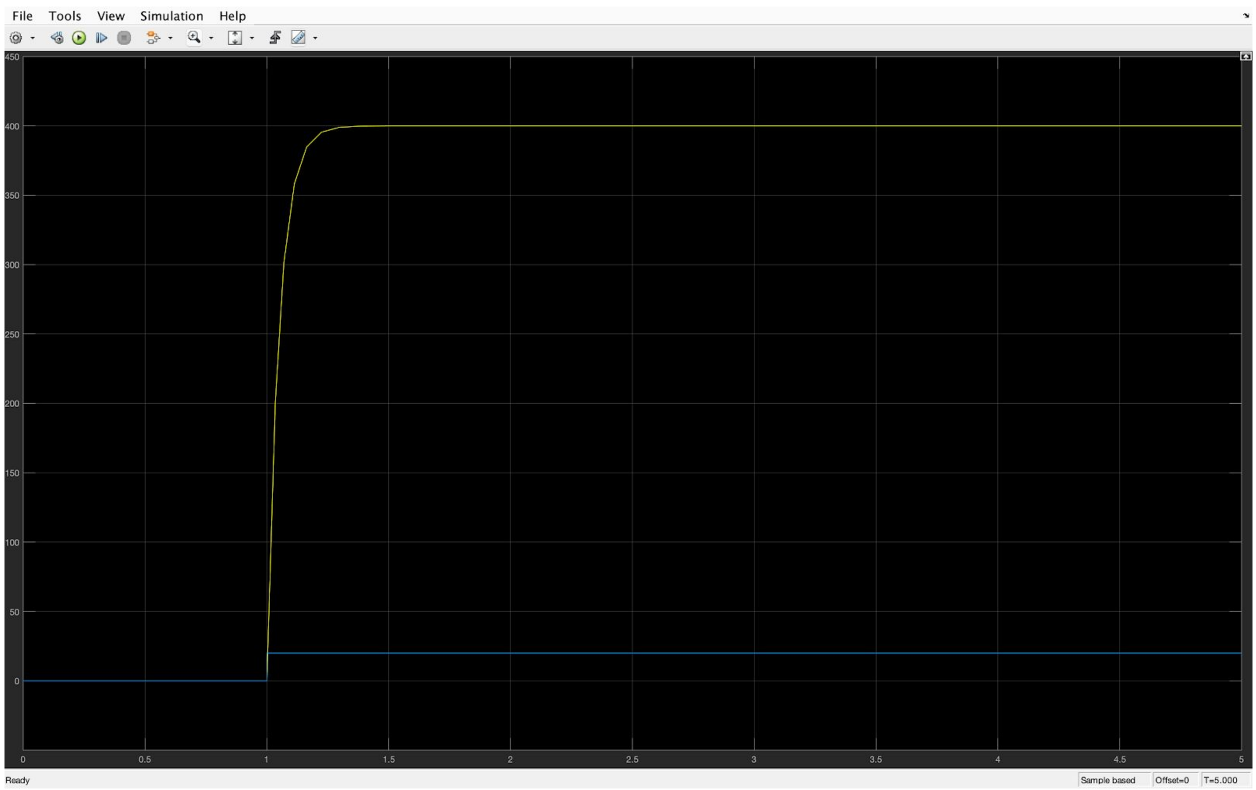

Figure 4. The expected output of the motor. Yellow is output, blue is input

Using this idea, the characteristic equation for this motor would look like:

$$G(s) = \frac{K}{S\tau+1}$$

$$K = \frac{output}{input} = static\ gain$$

$$\tau = T_2 - T_1;\ T_2 = time\ at\ Max * 0.63,\ T_1 = time\ at\ input$$

After doing running the test on our motor our characteristic equation resulted in:

$$G(s) = \frac{120}{S(0.0212)+1}$$

2.4 PID Tuning

The PID tuning process involves having a function that is able to approximate the motor, this equation was found in section 2.3. Using this equation we can tune the PID loop. Figure 5 shows the setup for PID tuning.

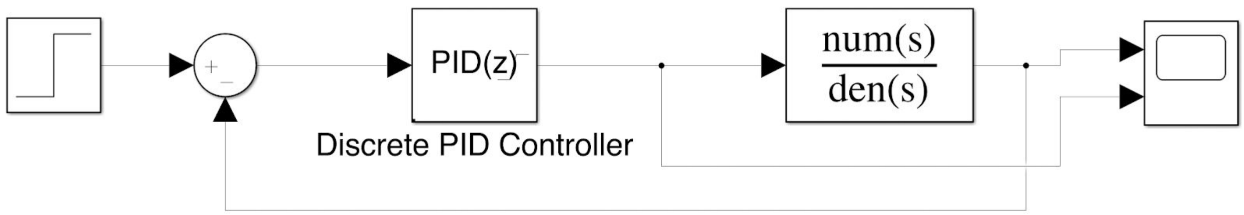

Figure 5. PID controller tuning setup.

After tuning the PID controller, we got the following coefficients for the controller and they are shown in Figure 6.

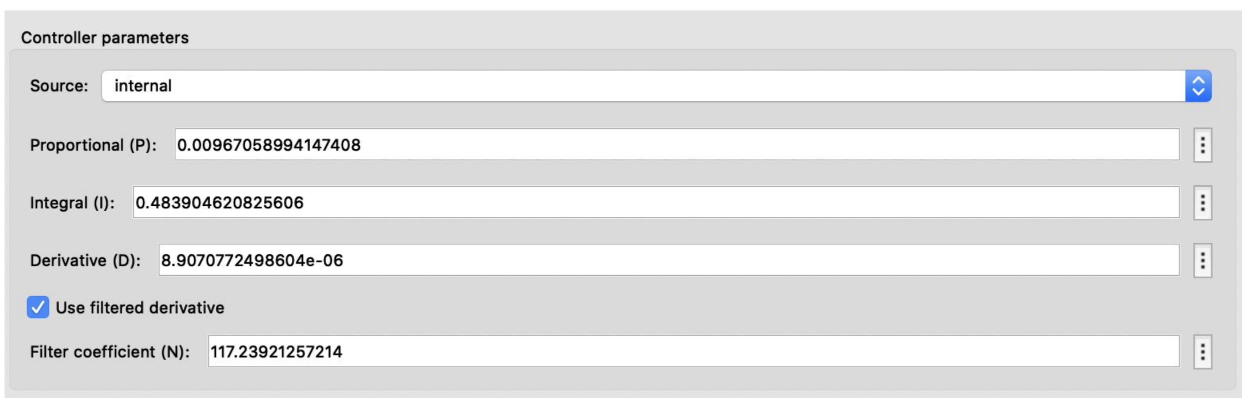

Figure 6. PID coefficients after tuning.

These coefficients were obtained by using the GUI sliders until the response satisfied the requirements that we were given.

2.5 Simulink Design

The final design replaces our discrete controller with the PID controller that we just tuned. We used our final design from Lab 3 to start our design. Figure 7 shows the final design of Lab 3.

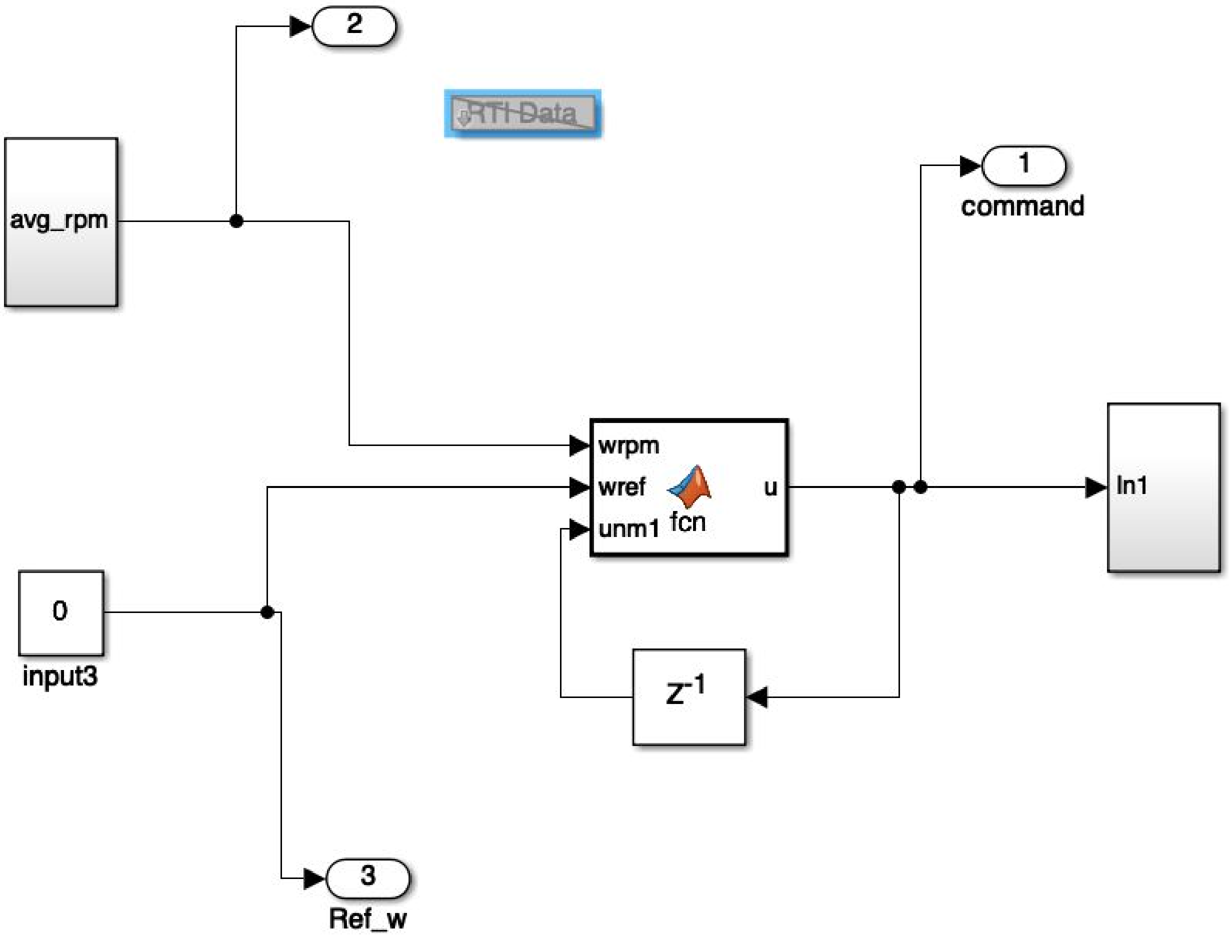

Figure 7. The final design of Lab 3 that we used as our starting point for Lab 7.

The final design for Lab 7 is shown in Figure 8. The new implementation features our tuned PID controller block.

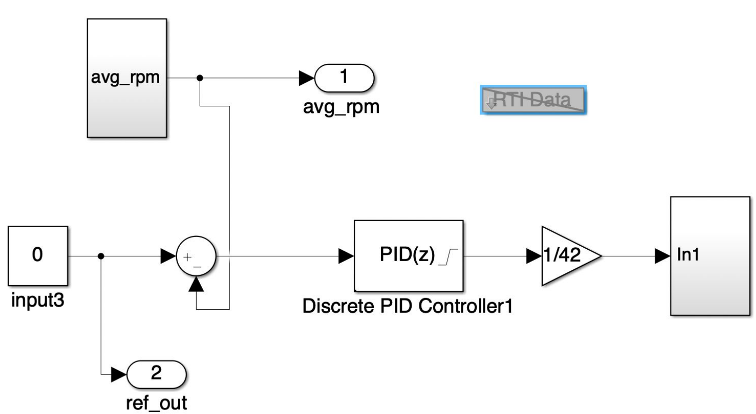

Figure 8. Lab 7 final design that implements a PID controller

3 Results

Similar to previous labs, we were required to tune the closed-loop PID to obtain a transient delay of less than 1 second, with a minimal steady state ripple. After experimenting with different transfer function coefficients to model a critically damped first order response, we were able to achieve the desired response as can be seen in the graph below where the rise time is ~0.1 seconds with a minimal voltage ripple for t > 0.1 after the step input.

Figure 9. The graphed output of the implementation of Lab7 final design.

5 Discussion

For this lab, we were able to successfully implement a first order transfer function for a dc machine. The response of the machine was controlled by a PID block in Simulink with coefficients determined by the program to achieve the desired damping effect and transient delay, in a manner to be within the required values. The addition of the PID demonstrated a significant improvement in the motor's response. The overall result was very similar to the

3-phase induction motor that was simulated in lab 6, but instead operating on dc principles. The implementation of the transfer function and associated PID block was very much the same as well, except modeled as a first order transfer function.